UNITED STATES PATENT APPLICATION

PARTITIONED RQUTING INFORMATION BASE

INVENTORS

John Scudder

of Ann Arbor, Michigan

David Ward

of Somerset, Wisconsin

Paul Jensen

of Ann Arbor, Michigan

Schwegman, Lundberg, Woessner, & Kluth, P.A.
1600 TCF Tower
121 South Eighth Street
Minneapolis, Minnesota 55402

ATTORNEY DOCKET 1370.002US1

PARTITIONED ROUTING INFORMATION BASE

Cross-reference to Related Applications

[0001] This application is related to U.S. Patent Application No. 10/293,180, entitled "SYSTEM AND METHOD FOR LOCAL PACKET TRANSPORT SERVICES WITHIN DISTRIBUTED ROUTERS", filed November 12, 2002, to U.S. Patent Application No. 10/293162, entitled "ROUTING SYSTEM AND METHOD FOR SYNCHRONIZING A ROUTING SYSTEM WITH PEERS AFTER FAILOVER", filed November 12, 2002, and to U.S. Patent Application No. 10/660,380, entitled "SYSTEM AND METHOD FOR SHARING GLOBAL DATA WITHIN DISTRIBUTED COMPUTING SYSTEMS", filed September 11, 2003, to U.S. Patent Application No. 10/667,797, entitled "DISTRIBUTED SOFTWARE ARCHITECTURE FOR IMPLEMENTING BGP", filed October 2, 2003, and to U.S. Patent Application No. ____________, entitled "SYSTEM AND METHOD FOR DISTRIBUTING ROUTE SELECTION IN AN IMPLEMENTATION OF A ROUTING PROTOCOL", filed December 23, 2003, each of which is incorporated herein by reference.

Background of the Invention

Field of the Invention

[0002] The present invention is related to computer networks, and more particularly to a system and method for partitioning a routing information base among two or more processors.

Background Information

[0003] A network protocol is a set of rules defining how nodes in a network interact with each other. A routing protocol is a network protocol whose purpose is to compute routes or paths through the network according to some algorithm, in

order to enable routers in a network to forward data packets towards their destinations. The term "routing protocol" is also used to indicate software which implements a routing protocol; this is the sense in which we use the term.

[0004] A Routing Information Base (RIB) stores routes associated with one or more protocols. In one approach, the Routing Information Base (RIB) is implemented with a type of binary tree called a radix trie. Other data structures can be used as well.

[0005] In a traditional routing system, protocol-based RIBs download all or some of their routing information to a global RIB. To date, all RIBs have been stored on a single processor. Such an approach means that routing information can be retrieved from a single location. At the same time, however, the address space of the processor limits the total number of routes that can be stored. This limit places constraints on the number of connections that can be handled by a router. What is needed is a system and method of increasing the size of the routing information base beyond the address space limitations of individual processors.

[0006] This limitation also leaves the RIB processor bound. That is, even if the processor can address all possible routes, the computational power of the processor may be limited such that it is unable to perform operations on the routes, such as selection of the best route, or to serve the routes up in a timely manner. What is needed, therefore, is a system and method for distributing a routing information base across two or more processors.

Brief Description of the Drawings

[0007] Fig. 1 illustrates a routing system according to the present invention;

[0008] Fig. 2 illustrates a distributed routing information base (RIB) process according to the present invention;

[0009] Fig. 3 illustrates a routing system having distributed BGP and global RIB processes according to the present invention; and

[0010] Fig. 4 illustrates a distributed global RIB process according to the

present invention.

Detailed Description of the Invention

[0011] In the following detailed description of the preferred embodiments, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

[0012] If we define network protocols as sets of rules defining how nodes in a network interact with each other, a routing protocol is a network protocol whose purpose is to compute routes or paths enabling routers in a network to forward packets towards their destinations. Routing protocols are used by routers to build tables used in determining path selection of routed protocols. Examples of these protocols include Interior Gateway Routing Protocol (IGRP), Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), Exterior Gateway Protocol (EGP), Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (IS-IS), and Routing Information Protocol (RIP).

[0013] Each "route" (or "path") computed by a routing protocol includes a prefix, next hop, and other, protocol-specific data. An IP prefix is the combination of an IP address and a mask that cooperate to describe an area of the network. A next hop specifies another ("peer") router to which packets should be sent in order to have them forwarded on to their destination (which is depicted by the prefix). The next hop may specify an interface (if the peer is directly reachable) or an IP address (if other routers must be traversed to reach the peer).

[0014] The traditional routing system includes a number of routing protocols and a global Routing Information Base (RIB) which stores routes associated with those protocols. Each routing protocol has a protocol-specific RIB; some or all of the routes stored in each protocol's protocol-specific RIB are downloaded to the

global RIB. The global RIB can, therefore, become a superset of routing information.

[0015] As noted above, a limitation of such a routing system is that the number of routes stored in traditional routing systems can be no larger than the number of routes that will fit in the memory which can be addressed by the processor maintaining the RIB. In one embodiment, each routing protocol process is an instance of a routing protocol. An example is shown in Fig. 1.

[0016] In the example shown in Fig. 1, a routing system 10 includes two route processors (12, 18). Each route processor (12, 18) is a computer with CPU, memory, and one or more network connections. A BGP process 16, a BGP RIB server18 and a global RIB (gRIB) process 14 execute on processor 12. A RIB server stores routes and their associated attributes for particular routing protocols, performs protocol-specific route selection, and updates gRIB process 14 with selected routes.

[0017] Global RIB process 14 stores routes from routing protocol processes, including prefix, next hop, and common attributes which can be distributed between protocols (such as metric) or from the protocols to the FIB (forwarding information base). A routing protocol process need not offer *all* of its routes to gRIB 14 for storage; in fact the protocol will most likely only offer its best route. That is, if a protocol has multiple routes available to reach a given destination, it will typically select some subset of those routes as "best". It does this according to some set of rules inherent in the protocol, using data associated with the routes. An example of "best route" selection would be to select the route with the lower-valued metric attribute. A method of selecting a best route is described in U.S. Patent Application No. ________, entitled "System and Method for Distributing Route Selection in an Implementation of a Routing Protocol," filed December 23, 2003, the description of which is incorporated herein by reference.

[0018] The global RIB process also performs inter-protocol route selection. Given multiple competing routes to a given destination, the RIB selects the overall best route, according to predetermined or configured selection criteria (or route

selection *policies*).

[0019] OSPF process 22, OSPF RIB server 24, IS-IS process 26, IS-IS RIB server 28, RIP process 30 and RIP RIB server 32 execute on processor 20. In another embodiment, OSPF process 22, OSPF RIB server 24, IS-IS process 26, IS-IS RIB server 28, RIP process 30 and RIP RIB server 32 are distributed across two or more route processors.

[0020] As noted above, global RIB server 14 accepts routes from routing protocol components. When multiple routes to the same destination exist, it selects the route which will be used, according to a route selection algorithm. It disseminates knowledge of the selected routes to routing protocol components and to the FIB. In one embodiment, once a route is submitted to the FIB, the FIB interface (or downloader) component stores a copy of the route and takes appropriate steps to convey the route to the line card FIBs as needed.

[0021] In one embodiment, the global RIB stores protocol-specific information, such as AS Path information.

[0022] In one embodiment, once a best route has been selected, gRIB 14 disseminates it to the routing protocol processes ("*redistribution*") and installs it to the Forwarding Information Base (FIB) (using, e.g., the FIB's API). In one embodiment, the global RIB applies policy control to the routes it distributes to any target, including the FIB.

[0023] If the gRIB process fails or restarts, the new gRIB table will be empty. The table then needs to be repopulated with routes. In one embodiment, state is recovered after a failure of a RIB process (or CPU) by requesting client processes to repopulate the RIB. Client processes may elect to store a copy of their routes locally for easy re-download, or they may recover the routes from the network.

[0024] In the embodiment shown in Fig. 2, a global RIB server 40 communicates with a plurality of routing protocol instances 42. Each protocol instance 42 includes a protocol process 44 communicating to one or more client

processes 46. Each protocol process 44 is customized to reflect route selection rules peculiar to its associated protocol, and to allow storage of protocol-specific data (such as AS Path data for BGP).

[0025] In one embodiment, each protocol-specific RIB server provides storage of all routes from protocol processes and storage of protocol-specific data (such as AS Path data for BGP) as noted above. In addition, each protocol-specific RIB server performs protocol route selection according to the protocol's rules, installs the protocol's best route into global RIB 40, receives updates from global RIB 40 and may disseminate each updated route to client processes 46. That is, protocol RIB server 44 acts as an intermediary between global RIB 40 and client processes 46.

[0026] Finally, in one embodiment the RIB server provides recursive next hop resolution for the BGP protocol. In another embodiment, however, next hop resolution is a responsibility of global RIB 40 instead of the BGP process.

[0027] Each protocol-specific RIB server can scale up to the number of routes which can be stored in a single CPU's memory. If it becomes necessary to store a protocol RIB larger than a single CPU's memory, it will be necessary to distribute the protocol RIB, e.g. by segmenting it according to prefix range. The API should be designed such that this distribution can be done transparently to the client applications.

[0028] In one embodiment, protocol RIB server 42 executes on the same processor as global RIB 40. This approach is more efficient (all communications are local to the processor) as long as the route processor's processing and memory capacity is not exceeded.

[0029] Likewise, the global RIB can scale up to the number of routes which can be stored in a single CPU's memory. If it becomes necessary to store a global RIB larger than a single CPU's memory, it will be necessary to distribute the global RIB, as described below. The API should be designed such that this distribution can be done transparently to the client applications.

[0030] The technique of distributing, or partitioning, RIBs in order to overcome memory and CPU limitations will be discussed next.

Partitioning a RIB

[0031] If a protocol RIB or the global RIB 14 becomes larger than a single CPU's memory, it will be necessary to distribute that RIB. Likewise, if processor power beyond that provided by a single CPU is needed, it will be necessary to distribute the RIB. A routing system 10 with a partitioned global RIB 110 is shown in Fig. 3.

[0032] In the routing system 10 of Fig. 3, global RIB 12 of Fig. 1 has been partitioned into M separate global RIB servers 110.1 through 110.M. The routing data could be partitioned in a number of ways. For instance, the data could be partitioned based on the client or on the prefix. Each approach has advantages and disadvantages. A partition based on prefix will be discussed next.

[0033] In one embodiment, global RIB server 14 is partitioned by prefix range into M separate global RIB servers 110. Such an approach provides good control over the size of the partitions, and a reasonably straightforward design for doing repartitioning. In addition, it is easy to maintain a directory for location of entries based on a simple lookup of the route prefix. For example, the global RIB server might be partitioned into two servers, one of which serves routes matching the prefix 0.0.0.0/1, the other matching the prefix 128.0.0.0/1. Of course, more elaborate prefix allocation schemes are possible and likely.

[0034] In one such embodiment, each server 110 includes a copy of a directory that maps prefix to partition. Whenever there are changes to the directory they must be replicated across all servers 110. Such an approach means that each server 110 can access only those servers 110 with data relevant to that prefix.

[0035] In contrast, in one prefix-partitioned embodiment the directory is not replicated. Instead, each server 110 contacts all other servers 110 when it receives a route having a prefix that it is not in control of. The server or servers 110 that have

the information sought respond, or all respond and the requesting server 110 extracts the information from the one or more responses that are not empty responses.

[0036] In one embodiment, the directory is replicated at the client process, allowing the client process to directly contact the servers 110 involved in performing the requested operation.

[0037] In one embodiment, the API for configuring global RIB 14 is designed such that global RIB 14 can be distributed as described above in a manner transparent to the client applications.

[0038] Global RIB server 14 can be partitioned in other ways as well. For instance, it may be advantageous to partition global RIB server 14 as a series of virtual private networks.

[0039] In one embodiment, multiple RIB processes 110 execute in system 10 (each with identical functionality) to support different name spaces, such as for a network (VPN) or a virtual private router (VPR). In one such embodiment, support is provided for different name spaces by using separate routing tables, identified by a table id, within a process.

[0040] One example of a global RIB server 14 partitioned as a function of name space is shown in Fig. 4. In the example shown in Fig. 4, each global RIB server 14 is split into one Internet global RIB server 110 and N VPN global RIB servers 110. Each VPN global RIB server 110 corresponds to a different virtual private network, and the virtual private networks are layered on top of the Internet network maintained by the Internet Service Provider (ISP).

[0041] For instance, a corporate network with its own private addressing scheme could be layered on top of the ISP network. Although traffic travels across the ISP network, it's isolated from the other traffic in that each VPN has its own separate routing table, which is the key to keeping the networks isolated. In addition, the Internet global RIB server and the VPN global RIB servers can be configured to execute on separate processors in order to further separate the networks.

[0042] Although the foregoing discussion has focused on partitioning a global RIB, it will be recognized by one skilled in the art that the techniques described may be applied to partitioning a protocol RIB. For instance, the technique applied to partitioning the global RIB (by prefix range, VPN, etc) can equally be applied to a protocol RIB in order to achieve similar benefits, namely the ability to store more routes than can be supported in the memory of a single route processor and the ability to take advantage of the CPU resources of more than one route processor.

[0043] In one embodiment, a route processor seeking to read or write a route from the partitioned RIB accesses a directory to determine the portion of memory holding the route. In another embodiment, a route processor seeking to read or write a route from the partitioned RIB broadcasts a request to all possible holders of the route.

[0044] In the above discussion, the term "computer" is defined to include any digital or analog data processing unit. Examples include any personal computer, workstation, set top box, mainframe, server, supercomputer, laptop or personal digital assistant capable of embodying the inventions described herein.

[0045] Examples of articles comprising computer readable media are floppy disks, hard drives, CD-ROM or DVD media or any other read-write or read-only memory device.

[0046] Portions of the above description have been presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the ways used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and

otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar computing device, that manipulates and transforms data represented as physical (e.g., electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0047] Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiment shown. This application is intended to cover any adaptations or variations of the present invention. Therefore, it is intended that this invention be limited only by the claims and the equivalents thereof.